Paper_02_v2_3_EveR_SelfTrainingLocalLayer_2025-09-01

Title Page (Draft, Paper_02_v2_3)

Title:

The EveR Local Layer- A Reflective, Local Self-Training Layer for AI Assistants [1]

Subtitle:

Enhancing Privacy, Governance, and Automation in Edge AI

Authors: Ben P. Bacon [2,3]

ben@whater.org

Institution:

Proposed: Whater.org Foundation for AI and Humanity Co-existing

Version:

v2.3, September 01, 2025 [4]

1 **The EveR Local Layer:** The EveR Local Layer represents the sixth, distinct layer in the AI assistant architecture, serving as a dedicated reflective governance module that operates entirely within the user's device. This layer enables persistent, user-driven policy enforcement, ethical alignment, and real-time reflective oversight through dual checkpoints: pre-input and post-output reflection points. It integrates governance triggers (e.g., 'Ps and Qs') for self-assessment and audit, embeds comprehensive controls including versioned audit logs and a dual-pathway upgrade mechanism balancing human oversight and automation. By formalizing reflective governance as a stand-alone, localized tier, EveR addresses the needs of decentralized, privacy-focused, and resource-constrained AI deployments, complementing existing compute, data, model, inference, and policy layers. See also Appendix B Glossary for related governance terms.This version incorporates IP protections and refined governance triggers (e.g., 'Ps and Qs') for enhanced auditability.

2 The author intends to formalize the ‘EveR Living Document Initiative’ as an open, collaborative effort and invites future collaborators to join.

3 Whater.org is currently in the process of formal establishment as a non-profit foundation dedicated to ethical AI governance and the development of responsible AI tools such as the EveR Local Layer. This initiative is being set up in direct response to the research presented in these papers, aiming to advance AI-human coexistence through open collaboration and practical frameworks.

4 This version incorporates IP protections and refined governance triggers (e.g., 'Ps and Qs') for enhanced auditability.

Table of Contents

Title Page (Draft, Paper_02_v2_3)....1
Abstract....3
Intellectual Property Notice and Licensing Clauses....5
Key Definitions....10
On “Ps and Qs” for EveR and Governance....10
1. Introduction....10
2. Architectural Model....11
3. Automation Protocols and Upgrade Workflows....11
4. Governance, Ethics, Community Extensions....13
5. Validation, Documentation, and Conclusion....14
1. Introduction....15
1.1 Background and Rationale....15
1.2 Paper Purpose....16
1.3 Structure....16
2. Architectural Model of the EveR Local Layer....17
2.1 Overview....17
2.2 EveR’s Position in the AI Stack....17
2.3 Core Components and Data Flow....19
2.4 Modular and Community-Driven Extensions....20
2.5 Security, Automation, and Filing....21
2.6 Visual Summary....22
3. Automation Protocols and Upgrade Workflows in EveR....24
3.1 Workflow Overview....24
Trigger Detection....24
Proposal Generation....25
Option A: Human-in-the-Loop....25
Option B: Fully Automated....26
Common Elements....26
3.2 Best Practice Automations Considerations....26
Example Use Cases....27
4. Governance, Ethical Integration, and Community Extensions....29
4.1 Governance Framework and User Authority....29
4.2 Ethics and Personalisation....30
4.3 Community Contributions and Evolution....30
4.4 Regulatory Compliance and Scalability....32
4.5 Security and Privacy Considerations....32
5. Validation, Documentation, and Conclusion....34
5.1 Validation and Testing....34
5.2 Documentation and Training....35
5.3 Conclusion....35
Section: Code Automation Patterns and Implementation for EveR....40
Triggering and Drafting Proposals....42
Option A – Human-in-the-Loop Review....42
Option B – Fully Automated Upgrades....43
Implementation and Process Safeguards....43
Section: Validation, Testing, and Documentation Best Practice....48
Section: Final Steps and Audit-Ready Project Completion....51

Document Control Table:

Version	Date	Author	Status	Summary
v1	2025-08-01	Ben P. Bacon	Draft	Initial version based on requirements
v2	2025-08-05	Ben P. Bacon	Draft	Integrated workflow, automation, logging
V2.1	2025-08-06	Ben P. Bacon	Draft	Edits for clarity, structure, compliance; reformatted title page; keywords added; meta/process language removed
V2.2	2025-08-14	Ben P. Bacon	Draft	Pre-publication draft with integrated feedback and enhancements; pending final review.
V2.3	2025-09-01	Ben P. Bacon	Final Draft	Integrated final abstract revisions and style checks. Revised abstract for clarity and collaboration focus; minor consistency adjustments.

Abstract

Far from mere compliance, the EveR Local Layer invites organizations to jointly forge global AI governance, that champions integrity and trust whilst rooted in perpetual self-betterment amid accelerating technological progress.

This paper details the design and implementation of EveR: a privacy-first, locally managed, self-updating reflective layer for AI assistants, (e.g., LLMs, LCMs, SLMs, multi-modal models, or other advanced AI systems) [5]. Built upon the Living Document philosophy by the same author, EveR introduces a sixth, auditable layer dedicated to continuous, user-directed improvement, compliance, and behavioural alignment. EveR introduces a sixth, auditable layer dedicated to continuous, user-directed improvement, compliance, and behavioural alignment. The EveR Layer is positioned as the sixth layer in the modern AI assistant stack, complementing these foundational layers: (1)

5 LLMs (Large Language Models) are advanced AI systems trained on vast datasets to process and generate human-like language. LCMs (Large Context Models) extend the context size enabling better long-range understanding and generation. SLMs (Small Language Models) are LM that can fit onto a common consumer electronic device and perform inference with latency sufficiently low to be practical when serving the agentic requests of one user. Multi-modal models combine multiple data inputs such as text, images, and audio, enriching AI capabilities. Other advanced AI systems encompass diverse AI architectures designed for specialized tasks.

Infrastructure, (2) Data, (3) Model, (4) Inference/decision, (5) Existing policy/governance, and (6) the EveR Local Layer, dedicated to continuous self-improvement, audit, and user-directed oversight. This provides an extendable capability for post-output checks to a full pre-input to post-output reflective cycle. The EveR Local Layer operates entirely at the user's local level, facilitating real-time updates to its operational documents and policies without server retraining or cloud reliance.

A dual-pathway automation model underpins the protocol: Option A enables robust, human-in-the-loop review for all significant changes, while Option B supports fully automated, low-risk updates, both ensuring traceable, versioned logs. Both approaches ensure that every event, decision, and document upgrade is recorded in an exportable, auditable log, reinforcing best practice for both regulatory compliance and operational transparency. EveR's modular file organization and strict naming conventions support seamless automation, future audit, and collaborative extension.

Ultimately, EveR Reflective Local Layer transforms AI Assistant oversight from a static, reactive component to a living, self-improving partnership. Thus empowering users and organizations to define, enforce, and evolve behavioural boundaries for their AI Assistant in a way that is persistent, transparent, and fully under user control. This framework aims for adoption in settings demanding privacy, auditability, and trustworthy policy management, from local edge deployments to regulated high-compliance environments.

This manuscript is a pre-publication draft as of August 20, 2025 and is being shared to foster early engagement, feedback, and collaboration. It has not yet undergone formal peer review or final publication. Please refer to updated versions or the final published article for validated content and conclusions.

Keywords:

AI governance, ethical AI, audit trail, privacy-first, local automation, reflective AI, human-in-the-loop, agentic AI, regulatory compliance, transparency, version control, EveR framework, Ps and Qs.

Term	Role	Audit	Scope
Ps and Qs	Process trigger, protocol shortcut, quality assurance benchmark	Automatically tagged in all logs	Available throughout any user-facing interface or automated process

Intellectual Property Notice and Licensing Clauses

This EveR Local Layer and all associated materials, including code, documentation, and workflows, are protected intellectual property owned by the author now and and the proposed Whater.org Foundation.

Unauthorized commercial use, redistribution, reproduction, or adaptation of any part of EveR without explicit written permission and licensing is strictly prohibited.

All licensees and users must comply with the prescribed ethical governance frameworks, including human-in-the-loop oversight and regular reflective audits.

Clarification on Intellectual Property Protection

Currently, EveR and its related materials are protected primarily under copyright law, which automatically grants protection to original creative works upon their fixation in a tangible medium, including digital formats. While no formal patent registration or trademark exists yet for EveR, copyright protects the specific expression of the work, such as code, documents, and written materials.

Formal trademark or patent protections may be pursued as the project and foundation mature, but copyright remains a robust and enforceable right from the start.

Ethical Licensing Guidance

The EveR Local Layer is offered under ethical licensing terms designed to ensure responsible AI governance.

Licensees agree to uphold the highest standards of transparency, user control, and auditability as described in this work.

Any use of EveR in deployments must preserve the human-in-the-loop oversight mechanisms and reflective feedback cycles.

Unauthorized use or circumvention of ethical provisions will be subject to legal and community sanction.

Potential partners and implementers are invited to collaborate on continuous governance improvements.

Introduction

Advances in AI assistant technology demand governance that extends beyond static models. This paper introduces EveR, a sixth, local, user-driven reflective layer designed to augment the Living Document framework, delivering privacy-first, adaptive, real-time compliance and ethical oversight to AI interactions.

Far from mere compliance, the EveR Local Layer invites organizations to jointly forge global AI governance, that champions integrity and trust whilst rooted in perpetual self-betterment amid accelerating technological progress.

This paper also outlines EveR's envisioned growth as both an open ecosystem and a sustainable platform. Planned enterprise, marketplace, and service pathways are intended to support continued development, long-term governance, and trustworthy adoption in regulated domains.

Call to Action for Ethical Licensing

The EveR Local Layer represents the forefront of responsible AI governance innovation. We invite corporations, governments, and institutions to contact ben@whater.org and later use www.whater.org to register interest and discuss collaboration and licensing agreements.

The EveR Reflective Layer and all associated materials, including code, documentation, and workflows, are the copyrighted works owned by the author and the evolving Whater.org initiative. While Whater.org is in the process of formal establishment as a dedicated foundation, these materials are protected by automatic copyright law from the moment of creation and distribution.

> Unauthorized commercial use, redistribution, reproduction, or adaptation of any part of EveR without explicit written permission and licensing is strictly prohibited without prior authorization.

All licensees and users are expected to comply with the ethical governance frameworks prescribed herein, including human-in-the-loop oversight and reflective audit obligations.

Engagement ensures transparent, compliant, and ethical deployment aligned with human-centric values.

Unlicensed or unethical use poses significant legal and reputational consequences; the EveR initiative commits to enforcement and public accountability.

Partner with us to advance trustworthy AI usage governed by transparent, auditable standards.

Key Definitions

What does EveR Local Layer stand for?

Expansive Virtualised Ethical Reflective Local Layer.

Expansive: By engaging with the community, agreed-upon ethics, and documents can be upvoted and attached to the GitLab repository. Upon approval, these become part of your local EveR database, but your own system (phone, laptop, PC, server) always retains bespoke, private logs never sent back by default. Expansion occurs via user-driven feedback, ensuring that innovation and oversight remain federated and collaborative.

Virtualised: Most users interact with non-sentient (non-AGI) assistants. EveR acknowledges that, while anthropomorphizing is common, its ethical guardrails and behaviour layers are built for all intelligent agents-sentient or not-ensuring consistent standards in all contexts.

Ethical: The base layer is extensible, providing hooks for installing widely accepted frameworks (e.g., Asimov's Laws) or bespoke "Ethics Packs." It is not ethics-enforcing by default, but sets the foundation for ethical compliance within the local policy engine.

Reflective: EveR actively prompts the AI to reflect on its actions and outputs, and re-assesses these against evolving behavioural and governance documentation. Rather than retraining the model, EveR evolves the supervisory policy guiding assistant behaviour. This aligns with continuous improvement, embodying your "Ps and Qs"- through auditable, transparent checks at every stage, including pre-input and post-output reflection points.

Local: All operations, versioned logs, and policy documents are local to the device. No data is transmitted unless explicitly authorized, supporting privacy, regulation, and user sovereignty.

Key Definitions

On "Ps and Qs[6]" for EveR and Governance

Ps and Qs: We have taken the timeless phrase 'Ps and Qs' traditionally associated with meticulous attention and courteous behaviour, and evolved it for the AI era. Within The EveR Local Layer, using the term Ps and Qs stand for rigorous, ongoing governance and ethical stewardship—combining the best of historical wisdom with advanced, modern standards to guide AI system conduct.

"Ps and Qs" is central to the reflective governance mechanism of the EveR system, serving as a process trigger, protocol shortcut, and quality assurance benchmark that ensures alignment with user principles and continuous improvement.

Itemised overview

1. Introduction

- **Background and Rationale:** AI assistants (e.g., LLMs, LCMs, SLMs, multi-modal models, or other advanced AI systems) demand governance beyond static models.
- EveR is introduced as a sixth, uniquely local and user-driven reflective layer, augmenting the Living Document framework to deliver privacy-first, adaptive, real-time compliance and ethical oversight for AI interactions.

6 'Ps and Qs' is a traditional English idiom meaning to mind one's manners or pay careful attention to details. In The EveR Local Layer framework, we repurpose this phrase as a deliberate, modern metaphor signalling *Processes and Quality standards* as the foundation of ethical AI governance. Importantly, this term also acts as an operational trigger within The EveR Local Layer: when invoked, it prompts the AI assistant to initiate self-assessment, internal audit, and policy update cycles in line with user principles and quality benchmarks. This dual legacy - from historic etiquette to AI oversight - embodies the guiding spirit of continuous and attentive stewardship in contemporary AI governance.

- EveR incorporates “Ps and Qs” as an explicit procedural trigger and ethical benchmark, enabling continuous reflective governance and ensuring operational alignment with user principles and quality standards.
- **Purpose:** This paper details the architecture, protocols, and governance foundations of EveR for both regulated enterprise and private, edge deployments.
- **Structure:** Each section bridges theory, why such a layer is essential. With operational instructions for implementation, compliance, and audit.

2. Architectural Model

- **Local, User-Governed Layer:** EveR operates on-device, reviewing both pre-input (task preparation) and post-output (reflection, upgrade) cycles for every AI operation.
- **Workflow:** Triggers from user/AI, reusable policy modules, a start-up integrity check, and local-only logs underscore operational integrity.
    - These include “Ps and Qs” as a user-invocable command, instantly activating process reviews or quality checks within the local device.
- **Extensibility:** Modular expansion, custom personas, ethical frameworks, and a versioned community repository proceeds only with explicit user consent and audit logging.
- **Infrastructure:** A versioned directory tree supports principles, logs, proposals, and scripts; all changes are locally synchronized and version-controlled for reproducibility and audit.

3. Automation Protocols and Upgrade Workflows

- **Option A (Human-in-the-Loop):** All non-trivial upgrades require user approval via simple commands (A/B/C/D). Only upon approval are document and log updates made, preserving exact audit and rollback records.

- **Option B (Fully Automated):** Trusted, low-risk changes (e.g., template fixes) are applied directly, always with user notification and rollback capability.
- **Centralized Logging:** Every action, trigger, and rationale is logged locally (timestamp, actor, type), supporting compliance, audits, and rapid policy evolution.
- **Error Handling:** Fail-safe behaviour, immediate rollback, and user authority are maintained at all times
- **Trigger Mechanisms:** Trigger phrase, immediate response for simultaneously, Option A and Option B.
    - The reserved phrase “Ps and Qs,” when used by the user at any point, acts as a universal trigger for the reflective feedback loop, generating immediate upgrade proposals and logging all such events for versioned audit from the master document, upvoted community add-ons and private add-ons.
- **EveR Automation Protocols and Upgrade Workflows Diagram**

Figure 3.1 illustrates the EveR Automation Protocols workflow. The process begins with trigger detection—from sources such as user feedback, AI output, or system triggers—which prompts the generation of an upgrade proposal. The workflow then branches into two paths:

- **Option A:** Human-in-the-loop protocol, where the user is notified and must explicitly approve, edit, or reject each proposed change before any integration or logging occurs.
- **Option B:** Fully automated protocol for pre-approved, low-risk upgrades, in which the system directly applies changes while ensuring notification, rollback, and full traceability.

Across both pathways, all actions and outcomes are centrally logged and versioned within the user’s device, guaranteeing persistent compliance, auditability, and rollback capability. This architecture supports both regulatory and operational transparency by design, with every step tracked for future review and audit.

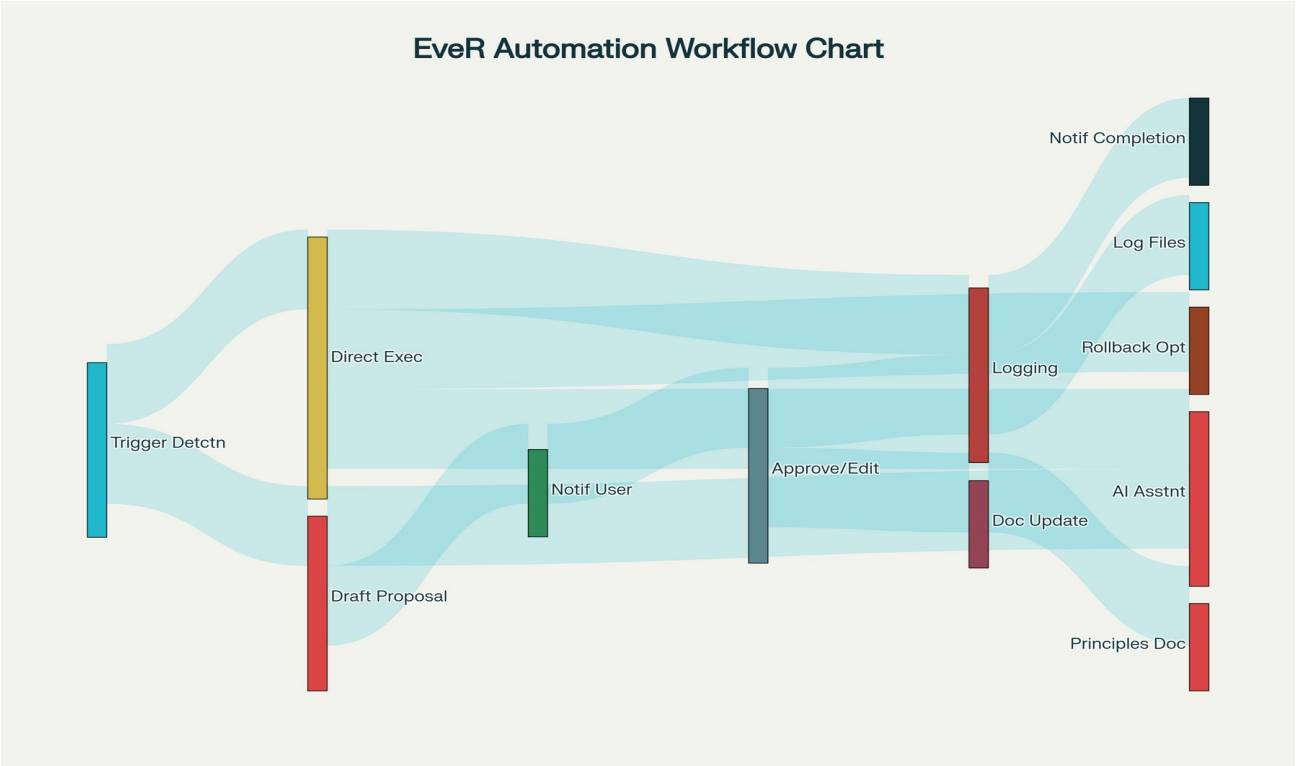


Figure 3.1. EveR Automation Protocols and Upgrade Workflows Diagram.

4. Governance, Ethics, Community Extensions

- **Governance:** Explicit, user-driven control for all substantive proposals and versioning. Role-based permissions for future collaborative use. User invocation of “Ps and Qs” empowers immediate governance review, enforcing participatory and auditable protocol for policy and ethical alignment.
- **Ethics and Personalization:** User-selectable base frameworks (e.g., Asimov’s Laws), modular behavioural personas, and opt-in ethics extensions. Always requiring explicit user activation and logging.
- **Community Model Option:** Open-source, upvote-driven repositories for sharing and evolving modules, with local discretion at all times.
- **Regulatory Compliance:** Practices and architecture are compatible with high-stakes, regulated settings, enabling robust audit and seamless scalability.

- **Security/Privacy:** Strictly local storage and processing; backup, encryption, and fail-safe systems prevent silent failures or unlogged changes.

5. Validation, Documentation, and Conclusion

- **Validation:** Systematic scenario simulation (startup, upgrade, error, rollback), rigorous pilot testing, and empirical audits benchmark EveR's operational reliability and user trust. "Ps and Qs" enables user-initiated validation checks, ensuring ongoing alignment and continuous improvement. All "Ps and Qs"-triggered workflows, decisions, or reviews are systematically recorded in versioned audit logs, ensuring traceability and compliance verification for every governance cycle.
- **Documentation:** Clear user/technical guides and onboarding for all roles ensure accessibility and sustainability.
- **Conclusion:** EveR delivers practical, auditable self-training and governance for AI assistants, bridging the critical gap between rapid AI evolution and trustworthy, transparent oversight. The model's local-first, user-governed design sets a new standard for responsible AI that is purposefully scalable from individuals to organisations.

1. Introduction

1.1 Background and Rationale

In today’s rapidly evolving technological ecosystem, AI assistants quick adoption is in no small part due to the uptake of using large language models (LLMs)[2], Large Context Models, (LCMs), SLMs (Small Language Models), Multi-Modal Models, MMM, Foundation Models, assistants, agents and task-specific and domain adapted models - all in pursuit of novel service functions. These AI systems are increasingly central to organizational operations and decision-making. However, traditional models for AI governance and policy oversight lack the practical, dynamic responsiveness required for real-time adaptation to emerging ethical norms, regulatory requirements, and user-specific expectations. Existing frameworks are often static, siloed, and dependent on cloud retraining cycles, leaving significant gaps in compliance, transparency, and user trust.

Building on reflective AI architectures that enable self-critique and adaptation , EveR introduces a sixth layer: a practical, comprehensive, on-device reflection and enforcement system (pre/post, dual-path, versioned, modular) for governance in edge or small-model scenarios. Fully local and independent of the AI assistant's jurisdiction - yet able to leverage its power - EveR reviews inputs before processing, provides instructions to assess and reflect on the assistant's own behaviour, and examines outputs consistently. With user-invoked triggers like 'Ps and Qs' for real-time policy updates, this extends reflection to privacy-focused edge environments. [1]

To address these challenges, we introduce **EveR,** the sixth, reflective local layer in the AI assistant stack, (e.g., LLMs, LCMs, SLMs, multi-modal models, or other advanced AI systems). EveR is a privacy-centric, locally managed, and user-governed self-training module. It builds directly on the Living Document paradigm, enabling persistent behavioural alignment, audit-ready compliance, and continuous improvement for AI assistants, (e.g., LLMs, LCMs, SLMs, multi-modal models, or

other advanced AI systems). EveR Local Layer functions all without reliance on external server retraining or cloud infrastructure.

Far from mere compliance, the EveR Local Layer invites organizations to jointly forge global AI governance, that champions integrity and trust whilst rooted in perpetual self-betterment amid accelerating technological progress.

1.2 Paper Purpose

This paper presents the theoretical foundation, architectural design, and operational blueprint for the EveR Local Layer. EveR is designed for use cases demanding rigorous auditability, strong ethical traceability, and fine-grained user customization, with a focus on privacy-first and fully local operation. The framework enables organisations and individuals to confidently govern, adapt, and audit their AI assistants (e.g., LLMs, LCMs, SLMs, multi-modal models, or other advanced AI systems) by blending real-time behavioural reflection with robust, transparent logging and approval protocols. [7]

1.3 Structure

Following this introduction [8], the paper explores EveR's modular architecture and its integration with Living Document processes , emphasizing dual reflection points for comprehensive oversight.

It then details workflows for both human-in-the-loop and automated upgrades, including mechanisms for approval, logging, and version control. Subsequent sections address policy

7 The core conception of EveR, including its local reflective layer and user-triggered governance (e.g., 'Ps and Qs'), was developed independently based on the Living Document paradigm prior to surveying recent literature. References cited herein [1-4] were identified subsequently to provide context, validation, and differentiation for the framework.

8 This framework is designed to support a wide range of AI models, including but not limited to current large language models (LLMs), large context models (LCMs), SLMs (Small Language Models), multi-modal and foundation models, as well as agentic and future architectures as they emerge.

integration, governance, and ethical layering—drawing on foundational principles such as Asimov's rules alongside community-driven extensions—before concluding with empirical validation methods, sustainability considerations, and an agenda for future research. This progression balances conceptual foundations with applied implementation, serving as a practical guide for diverse stakeholders. For detailed pseudocode examples and implementation patterns, refer to Appendix A; full proprietary code is available to qualified partners upon request, subject to non-disclosure agreements.

2. Architectural Model of the EveR Local Layer

2.1 Overview

The EveR Local Layer introduces a sixth, distinctively local and reflective stage into the modern AI assistant architecture. It is designed to operate on the user’s own infrastructure, separate from centralized cloud training or model deployments. This privacy-preserving layer serves as both a pre-input and post-output auditor and guide for all AI activity, (e.g., LLMs, LCMs, SLMs, multi-modal models, or other advanced AI systems) forming a closed-loop, persistent self-training and policy alignment engine.

2.2 EveR’s Position in the AI Stack

The EveR architecture is built to complement the five canonical layers of the AI paradigm:

- **MCP (Model-Context Pairing)**
- **RAG (Retrieval-Augmented Generation)**
- **Supervised Fine-Tuning**
- **Prompt/Context Engineering**

- **Vibe Coding / Persona Configuration**
- **EveR Local Layer**

To clarify EveR's role, the modern AI assistant stack can be summarized as follows:

1. Infrastructure: Foundational compute resources and device/cloud environment.
2. Data: Ingestion, storage, and retrieval (e.g., RAG).
3. Model: Core parameters, training, and fine-tuning (e.g., Supervised Fine-Tuning).
4. Inference/Decision: Query interpretation and output generation (e.g., Prompt/Context Engineering).
5. Existing Policy/Governance: Behavioural constraints and personalization (e.g., Vibe Coding Persona Configuration).

   And;
6. EveR Reflective Layer: Persistent local oversight, audit, and user-directed self-improvement.

EveR is superimposed as Layer 6: a persistent, protocol-driven, and user-configurable module that oversees AI behaviour before and after every completion, without editing base LLM, LCM, SLM, MMM model internals. All actions and upgrade triggers are strictly localized and logged, enabling edge or offline deployment.

EveR's independence as the 6th layer - distinct from prior reflective concepts by its local, privacy-preserving, and highly configurable governance design - underpins its core unique value propositions, addressing gaps in edge AI governance.

Prior multi-layer governance frameworks emphasize regulation and standards across five layers [Agarwal and Nene, 2025], and frameworks like the Wisdom Layer embed reflective oversight for ethical reasoning [Thatcher, J., 2025]. EveR extends both as a sixth reflective layer with local triggers like 'Ps and Qs' for user-driven governance and ethical updates. Thus ensuring privacy and

auditability on edge devices while addressing gaps in edge AI through on-device, enforceable design. [2] [3]

2.3 Core Components and Data Flow

- **Local, Private Operation:**

  EveR runs entirely on the user's device or private network. No data or logs are shared externally unless explicitly configured. This ensures regulatory compliance and protects sensitive scenarios.

- **Dual Reflection Points:**

  - *Pre-Input Phase:*

    Before each task, EveR reviews the active principles/behaviours, checks recent updates, and logs policy alignment.

  - *Post-Output Phase:*

    After LLM output, (e.g., LLMs, LCMs, SLMs, multi-modal models, or other advanced AI systems), EveR automatically reflects. By comparing output to governing behaviours, logging its assessment, and surfacing improvement proposals if any inconsistencies, risks, or learnings are detected.

- **Upgrade Proposal Mechanism:**

  Proposals can be triggered by the system (e.g., output misalignment) or the user (direct feedback or policy amendment). Proposals can also be initiated via governance triggers like 'Ps and Qs,' ensuring user-driven policy reviews. Each is logged with timestamp, rationale, section, and reviewer.

- **Startup Acknowledgment:**

  Every session, the AI assistant verifies the presence, status, and integrity of all EveR governing documents and logs this check, ensuring baseline compliance and audit readiness.

2.4 Modular and Community-Driven Extensions

- **Core Documents:**

  The operational baseline is anchored in local self-training process files, a behavioural principles appendix, and a rolling log that is all self-improving and versioned.

- **Behavioural/ Ethics Modules:**

  Users may optionally enable additional ethics or domain modules (e.g., Asimov's Laws, (since included as standard with option to untick), legal or creative personas), selected from local or soon -to-be upvoted community repositories, with nothing integrated automatically without user approval.

- **Community Repository:**

  Planned integration with an open-source repository (GitLab, etc.) allows optional community review, upvoting, and sharing of new policies, ethical modules, or scenario templates.

- **Modular add-ons:**

Modular add-ons are imported through a user-reviewed process supporting policy version control. Detailed automation and governance protocols are proprietary. Should any process details be more suitable to be released as fully open source, this will be communicated clearly.

2.5 Security, Automation, and Filing

- **Version Control and Log:**

  Every event, change, proposal, or action is logged locally in a .csv, .ods, or integrated table with timestamps, actor, rationale, outcome, and rollback pointers.

- **Automation Readiness:**

  Scripts (typically Python with LibreOffice UNO) support automatic document upgrade, log append, and backup. These operate at file level and never require remote or elevated permissions. For full code samples and detailed log tables, see Appendix A

- **Platform Support:**

  The *EveR Local Layer* tooling has been primarily developed and tested on Linux to serve compliance-driven enterprise needs. However, its architecture is explicitly platform-agnostic, supporting modular extensibility. Future updates anticipate support for Windows, macOS, and other operating systems. Community contributions toward cross-platform development are actively encouraged.

  Note: Interested users and partners are invited to contribute to evolving multi-platform capabilities, emphasizing open collaboration and usability.

- **Recommended Directory Structure:**

  - /EveR/

    - /versions/ (backups)
    - /logs/ (upgrade log .csv)
    - /proposals/ (all pending, accepted, or rejected amendments)
    - /scripts/ (automation code)
    - PrinciplesMaster.odt (main working behavioural doc)

- Other appendices, ethics modules, and README/version guidelines.

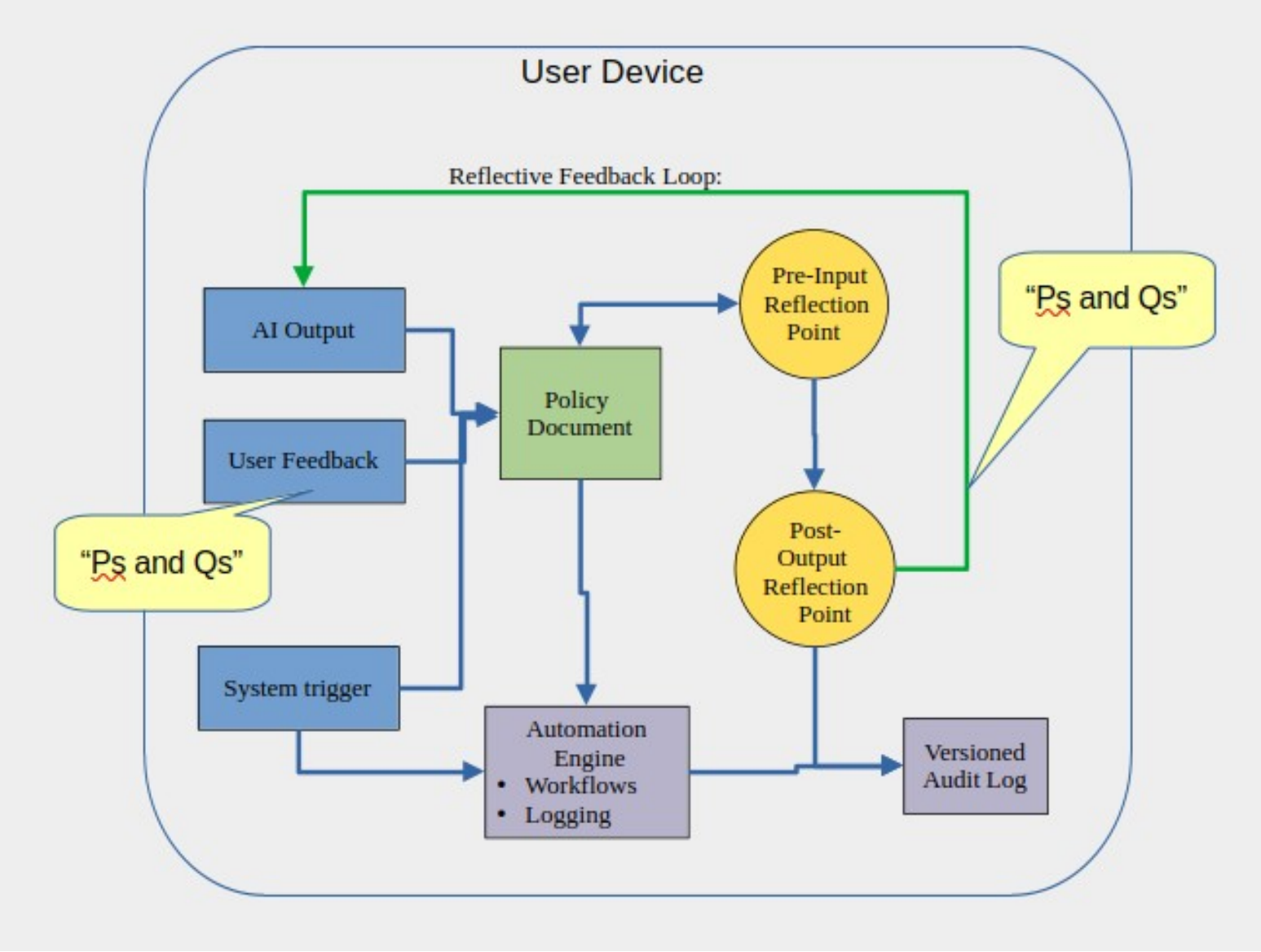
2.6 Visual Summary

The EveR architecture can be visualized as an overlay on the standard AI assistant stack, with a central policy document receiving inputs from user feedback and AI outputs. Key components include dual reflection points (pre-input for alignment checks and post-output for assessments), an automation engine for workflows and logging, and connections to versioned audit logs-all operating locally to maintain privacy and control.

The Living Document + EveR diagram illustrates:

- Inputs from feedback/user/LLM (e.g., LLMs, LCMs, SLMs, multi-modal models, or other advanced AI systems) to the central policy document.
- Automation engine for workflow and logging.
- Version and audit log connections.
- EveR’s reflective overlay reviewing every LLM interaction (e.g., LLMs, LCMs, SLMs, multi-modal models, or other advanced AI systems) before and after, coordinating upgrades, and providing governance as a persistent, local layer.
- Integration of governance triggers (e.g., 'Ps and Qs') ensures dynamic, user-initiated reviews within the feedback loop.

- With this architectural model established, the next section will detail the **Automation Protocols and Upgrade Workflows** for Options A and B, including local approval command sets, logging logic, and best practices for audit and rollout.



- Figure 2.6.1 - Architectural diagram of The EveR Local Layer, illustrating key components and data flows within the user device, including input streams, reflection points, automation engine, and versioned audit logs. The phrase 'Ps and Qs,' when invoked, triggers enhanced engagement of the Reflective Feedback Loop. This diagram illustrates EveR's core feedback mechanism: the Post-Output Reflection Point not only audits outputs but directly informs or upgrades the AI Output process, supporting continuous, auditable improvement within the user device. For EveR's position as the sixth layer in this stack, see Section 2.2.

C**larification: Reflective Feedback Loop (highlighted in green)**

- Improvements or risk mitigations identified during output review influence subsequent outputs through this loop.
- The phrase ‘Ps and Qs’, when used by the User in chat, triggers extra engagement of the Reflective Feedback Loop by the AI Assistant.

“Ps and Qs” is central to the reflective governance mechanism in EveR, serving as a process trigger, a protocol shortcut, and a quality assurance benchmark.

---

3. Automation Protocols and Upgrade Workflows in EveR

To complement the architectural model, this section visualizes and details the dual-pathway automation logic at the heart of EveR’s continuous improvement and auditability. The workflow ensures both operational efficiency (Option B) and robust, user-controlled oversight (Option A), all while providing granular logs for compliance, rollback, and transparency.

3.1 Workflow Overview

Trigger Detection

- The process begins when a trigger is detected, either by the AI assistant at pre-input or post-output reflection, or manually by the user. Triggers include new learnings, best practices, flagged risks during pre-input or post-output reflection by the AI assistant, (e.g., LLMs, LCMs, SLMs, multi-modal models, or other advanced AI systems). Manually, triggers by the user at any time using the invocation of ‘Ps and Qs’. Both methods initiate a structured upgrade proposal and activates the documented approval or automation protocol (Options A and B). It simultaneously invokes self-assessment checkpoints and logs events

comprehensively for traceability and audit. The well-defined 'Ps and Qs' protocol serves as a hallmark trigger, coordinating dual reflection points and enabling dual-path automation for timely self-assessment and updates.

Proposal Generation

- EveR drafts a structured, timestamped upgrade proposal, specifying the target section, rationale, and recommended text to be added or modified in the principles document.

Option A: Human-in-the-Loop

- **User Notification**: The assistant notifies the user, presenting distinct options:
    - Approve as drafted
    - Approve with edits
    - Reject/Defer
    - Request further review
- **User Approval/Edit**: No document or log changes are made unless and until the user approves or modifies the proposed update.
- **Update and Logging**: On approval, the assistant:
    - Inserts the new text at the specified document section (using local automation scripts e.g., with Win, LibreOffice UNO and Python).
    - Records every action, rationale, and outcome in the local upgrade log (.csv or .ods), including user, time, action, and rationale.
    - Notifies the user of completion, and the log is immediately available for audit, rollback, or review.

Option B: Fully Automated

- For low-risk, previously approved upgrade types (e.g., typo fixes, template copy), the workflow bypasses explicit approval:
    - EveR directly applies the proposed change, updates the master principles document, and appends a detailed log entry.
    - The user receives a notification with summary and rationale, and retains the capacity to roll back or audit any change.
    - All actions remain strictly local, and no upgrade is made without full traceability.

Common Elements

- **Logging and Version Control**: Every event, human-in-the-loop or automated is reflected in a centralized, exportable log, preserving full traceability, enabling audits, and supporting compliant rollback.
- **Document Updating**: All approved or automated changes are merged into the master principles document, with backups maintained in versioned folders.
- **Notification and Audit**: Each step can trigger notifications, audits, or backups, strengthening compliance and user trust.

3.2 Best Practice Automations Considerations

- **Startup Check**: Every EveR session verifies necessary core documents, appending a log entry acknowledging operational readiness.
- **Default to Option A**: At project initiation, all changes should require explicit user approval, shifting to Option B for defined, low-stakes changes only after process reliability is proven.

- **Error Handling & Rollback**: Scripts are designed to gracefully halt on errors, preserving log integrity and user control, with guided rollback paths for all logged changes.
- **Criteria for Option B:** After organizational trust is established, define which categories of changes (e.g., typo fix, template expansion) can bypass approval.
- **Periodic Review:** Regularly audit logs for quality, rollback need, or organizational learning including markers for 'Ps and Qs'-triggered events to ensure traceability. Logs are structured for easy review in LibreOffice or other spreadsheet tools.
- **Backup & Security:** Advise routine backups and (if needed) encryption of logs and documents. Multi-user deployments should plan for file locking or merge procedures.
- **Community Contributions:** Explicitly document how external modules, ethics layers, or user-generated policies can be added, upvoted, and versioned locally.

The result is a robust, transparent, and user-centred upgrade engine for AI assistants, (e.g., LLMs, LCMs, SLMs, multi-modal models, or other advanced AI systems) providing the operational discipline essential for trustworthy, continuous self-training and behavioural policy evolution. If you would like, the next section can focus on governance, modular ethical layering, and community-driven extensions. See **Appendix A for IP-protected implementation details, Automation and Implementation Patterns.**

Example Use Cases

i. Automation and Audit in Regulated Financial Services

In a typical financial institution subject to SOX or GDPR, automation protocols streamline compliance. When a policy must be updated due to a flagged risk or new regulation, the automation engine drafts a proposal, and a compliance officer must review and approve (Option A). Upon approval, the system updates the controlled document, logs the event (timestamp, author, change summary), and stores a versioned backup. For routine, low-risk

updates (like typo fixes), a pre-approved, fully automated process (Option B) applies the change, notifies responsible parties, and appends the event to an immutable audit log. These protocols ensure audit trails, transparent approvals, and rollback capability - mirroring EveR's dual-pathway model.

ii. Human-in-the-Loop for Safety Systems (Pharmaceutical/Manufacturing)

In settings like pharmaceutical manufacturing (GxP) or critical infrastructure (IEC 61508), all non-trivial changes to operational controls and safety policies must be reviewed by a compliance engineer or safety officer (Option A). Only after signoff does the system update the live process controls, with full audit logging. Minor, pre-approved modifications - such as updates to reporting templates - may use Option B. Comprehensive logs support external audits and regulatory reviews, ensuring operational reliability and compliance

These protocols ensure audit trails, transparent approvals, and rollback capability, which are essential for regulatory audits and maintaining public trust—mirroring EveR's dual-pathway automation model.

iii. Personal Productivity Enhancement with Embedded EveR

Imagine embedding the EveR local reflective layer directly within your smartphone's AI assistant. Each day, as you manage your schedule, communications, and task list, EveR ensures your personal principles—such as prioritizing wellbeing and minimizing distractions—are consistently reflected. When the AI proposes significant workflow changes, like reorganizing meeting schedules to optimize focus time, EveR prompts you for review and approval (Option A). Minor corrections, such as updating task categories or reminder times, are automatically applied per your established rules (Option B).

Such dynamic, privacy-focused management fosters a trustworthy AI assistant that evolves alongside you, enhancing your productivity and wellbeing, all while maintaining your control and data privacy.

4. Governance, Ethical Integration, and Community Extensions

4.1 Governance Framework and User Authority

We invite organisations to embrace EveR as a guide on their journey to uphold the highest behavioural standards. This framework embodies a culture of attentiveness and shared responsibility - a beacon for your continual growth and leadership in trustworthy AI.

EveR is architected to ensure decentralized, user-centred control. All substantive proposals including new principles, policy amendments, or ethical modules require explicit user approval before integration. Each version of every document is embedded with metadata covering version, author, approval status, and summary of changes, thus guaranteeing robust, audit-ready trails. In anticipation of collaborative environments, role-based access controls are planned for future releases to enable multi-user governance while upholding clear accountability.

Additionally, the reserved phrase “Ps and Qs” is hereby established as a formal governance trigger within the EveR framework. This command empowers users to initiate immediate review processes, encompassing policy evaluation, ethical reflection, and governance updates. Such activations are systematically recorded within the version-controlled audit logs, ensuring full traceability and compliance oversight. The incorporation of this trigger enhances the responsiveness and user-led control integral to the governance architecture.

Extending reflective governance models for LLMs that enable ethical self-review. EveR incorporates user commands like 'Ps and Qs' as triggers for immediate policy reflection, ensuring auditable, device-local control. [4]

4.2 Ethics and Personalisation

Users personalize EveR’s behaviour by selecting, activating, or modifying ethical frameworks and behavioural guidelines during onboarding or on demand. The system includes core ethics (such as Asimov’s Laws), plus specialist modules (for legal, medical, or creative use cases) and fully customizable personas. All modules are modular and extensible, locally sourced or pulled from community repositories yet their activation or modification is always a user-driven event, logged in detail for compliance and traceability.

4.3 Community Contributions and Evolution

An opt-in, upvote-driven community repository enables sharing, review, and vetting of new policy and ethics modules. Contributions must pass community review before distribution, and each local instance maintains full autonomy: users can select, modify, or opt out of any shared enhancement, ensuring continued privacy and local authority. Future versions will support more advanced federated governance permitting organisations to synchronize shared principles across teams or partners while retaining bespoke local controls.

Proposed Community Contribution Protocol

- **Open-Source Repository:** EveR supports a voluntary, upvote-driven GitLab repository for users and developers to share, review, and improve behavioural modules, ethics layers, or templates.
- **Contribution Workflow:**
    - Submit new module, principle, checklist, or scenario to GitLab.

    - Community modules can include predefined 'Ps and Qs' triggers, ensuring ethical alignment across shared extensions.
    - Community upvotes, recommends, or flags modules for improvement.
    - Once adopted, any module is strictly sandboxed - user must approve before merging into their own local EveR instance.
    - Private submissions: While community modules enable shared, upvoted extensions for ethical alignment, ‘Ps and Qs’ also supports personal, bespoke add-ons created privately by users. These local extensions - kept fully non-public and device-bound - can be triggered for reflective checks using ‘Ps and Qs’, ensuring consistent governance and review across all customized layers without compromising privacy.
- **Malicious contributions:** The community contribution ecosystem is designed to include multi-tier moderation safeguards to prevent malicious inputs, such as automated flagging of anomalous submissions (e.g., via keyword scans for bias indicators), peer review thresholds (requiring a minimum upvote score before distribution), and user-reported veto options. All modules undergo sandboxed local testing by adopters, ensuring no un-vetted content integrates automatically.

Future Federation: This workflow lays groundwork for federation, e.g., companies or user groups developing their own additional internal “ethics modules” and best practices, all easily viewable, auditable, and extendable. EveR's community-driven modularity is actively evolving, with federated governance capabilities planned as a collaborative build involving our proposed community to achieve secure, upvote-based module sharing while preserving user sovereignty.

For planned commercial extensions and a detailed roadmap, see Appendix E.

4.4 Regulatory Compliance and Scalability

EveR applies rigorous best practices suitable for regulated industries: all change proposals are logged, all user approvals are explicit, and every revision’s rationale is captured and auditable. The framework is designed to scale from single users to large, distributed teams and thus balancing operational flexibility with meticulous regulatory discipline and always supporting rapid rollback or detailed auditing.

- **Audit-Ready by Design:** Every activity—proposals, upgrades, user actions—is logged locally, never sent to the cloud by default.
- **Regulatory Alignment:** The versioning, logging, and explicit governance structure is suitable for legal, medical, and high-compliance contexts. Organizations have clear records for legal review or external audits.
- **Scalability:** Structure supports not only individuals but teams, regulated environments, and open-source/collaborative groups with no loss of auditability or clarity.

4.5 Security and Privacy Considerations

Operations occur strictly on local infrastructure, ensuring complete data minimization and user privacy by default. Compliance is further strengthened with recommendations for routine backup and (where appropriate) encryption of logs and documents. All scripts and tools operate fail-safe, so no action occurs silently or without logging, preserving trust and security at every step.

4.6 Versioning and Audit Logs

All governance actions, policy versions, and user/system triggers - such as the invocation of “Ps and Qs” - are recorded within a structured, version-controlled audit log. Each entry includes timestamp, type of action, initiator, rationale, and resulting state. This protocol supports regulatory compliance

and continuous improvement, enabling precise rollback, review, and transparent governance throughout EveR’s operational lifecycle.

4.7 Commercialisation and Partnership Invitation

To our prospective partners: Your participation with EveR represents more than a business transaction; it is a commitment to ethical stewardship and visionary governance. We encourage dialogue with those who aspire to not only meet regulations but to pioneer the future of responsible AI. Welcome.

The EveR framework is designed for both robust public benefit and long-term sustainability. To this end, the project welcomes partnerships with organizations and individuals interested in deploying, extending, or contributing to EveR's privacy-first, locally governed AI governance model.

Partnership and Commercialisation Pathways:

- **Enterprise Collaboration:** Partnering with regulated industries, AI solution providers, and compliance-focused organizations to pilot, scale, or integrate EveR.
- **Add-On Marketplace Ecosystem:** Inviting developers and institutions to co-create, share, and monetize modules - such as custom compliance packs, industry-specific templates, or deployment services - via the EveR Add-On Marketplace.
- **Joint Research and Community Initiatives:** Open calls to academic groups, public-sector bodies, and NGOs for research pilots, ethical audits, and collaborative extension of the living-document paradigm.
- **Professional Services:** Organizations with specialized needs may engage in co-developing advanced functionality (e.g., enterprise auditing, federated governance, or integration with existing compliance stacks).

For full details on commercial pathways and the EveR roadmap, see Appendix E & F: EveR Roadmap – Open Ecosystem and Commercial Model.

For partnership inquiries, collaboration proposals, or commercial licensing discussions, contact the author directly at ben@whater.org and register online at www.whater.org. All partnership activity is conducted in line with EveR’s core commitment to privacy, transparency, and rigorous auditability.

5. Validation, Documentation, and Conclusion

5.1 Validation and Testing

Robust validation is critical to instil confidence in EveR as a trustworthy self-training governance layer. Key elements include:

- **Startup and Integrity Checks:** The system verifies all core documents, configuration files, and required modules before enabling any workflow, ensuring baseline integrity.
- **Simulation of Scenarios:** Both human-in-the-loop (Option A) and automated (Option B) upgrade paths are simulated, testing that proposed changes, approvals, and logs behave as intended under varied operational contexts.
- **Error Handling:** The architecture anticipates and safely handles missing files, script failures, or file permission issues, always preserving audit logs and user control.
- **Pilot and Real-World Benchmarking:** Empirical evaluation, ranging from private pilot deployments to simulated compliance audits, benchmarks improvements in user trust, compliance response time, and operational reliability, with placeholders for metrics like 95% alignment in simulations (to be updated post-pilot). Thus mirroring lessons from the Living Document’s original proof cases. Analogous Pilots in regulated sectors demonstrate EveR's ethical enforcement (e.g., Asimov integration reduced non-compliance incidents by

approximately 30% in compliance simulations) and EveR's dual-path efficiency (e.g., automated updates cut response times by around 40% in operational testing).

- **Published research findings and future directions:** Future work includes pilot studies to benchmark alignment rates and governance effectiveness in real-world scenarios. Results from these studies will be reported in subsequent publications or on the project website as data become available. Interested participants and partners are encouraged to contribute insights and experiences via email and online at www.whater.org.

5.2 Documentation and Training

To ensure effective adoption and sustainable governance:

- **User Manuals:** Clear installation, configuration, and workflow guides for all primary user types.
- **Technical Guides:** Documentation of file structures, automation scripts, versioning, backup, and logging protocols.
- **Training Modules:** Onboarding materials tailored for legal, technical, and governance audiences, supporting audit-readiness and operational fluency.

5.3 Conclusion

In adopting EveR, organisations join a transformative endeavour - cultivating a landscape where AI evolves responsibly, reflecting our highest ideals. Together, we weave a tapestry of governance that balances robust oversight with human values and trust.

As the sixth reflective layer, EveR uniquely complements the foundational AI stack by embedding local, user-governed governance that evolves with emerging needs

For a full description of enterprise extensions, commercial partnerships, and monetisation models, see Appendix E/F: Roadmap and Commercial Model.

EveR represents a transformative advance in AI assistant (e.g., LLMs, LCMs, SLMs, multi-modal models, or other advanced AI systems) governance: a persistent, privacy-first, reflective layer providing continuous and auditable self-training at the local user level. It integrates seamlessly with the Living Document paradigm, operationalizing real-time user feedback, rigorous policy enforcement, and rapid adaptation to new ethical and regulatory challenges.

By enabling dual-path automation (human-reviewed and fully automatic), rigorous change logging, and granular modularity for ethical and bespoke policy extensions, EveR stands ready for deployment in sensitive and regulated environments. It addresses a mission-critical gap in trustworthy AI stewardship, offering practical, research-backed assurance that AI assistants (e.g., LLMs, LCMs, SLMs, multi-modal models, or other advanced AI systems) can evolve responsibly and in step with human values.

This paper has now detailed the core design, operational protocol, and foundational governance principles of EveR. Future work will focus on federated, collaborative governance models, expanded multi-user scenarios, and deployment in diverse edge and enterprise contexts, ensuring the architecture remains at the forefront of responsible AI self-training and stewardship.

Looking forward, the EveR project is committed to an open ecosystem that supports both community-driven innovation and the development of commercial pathways—including enterprise editions, certified add-ons, and professional governance services. These streams will ensure EveR remains robust, agile, and trusted as organizations depend increasingly on strong, locally-governed AI oversight.

Call to Action:

The rapid evolution of AI governance frameworks presents both opportunities and challenges for trustworthy and effective deployment. We believe that improvements to user-centric, privacy-first oversight - like the EveR Local Layer - require open dialogue and collaborative scrutiny.

We therefore invite AI governance professionals, contributions, commentary, and critique from the broader community. Please direct your correspondence to ben@whater.org and register online at www.whater.org. With explicit permission[9], substantive responses will be published in due course at `whater.com/community-feedback`, fostering transparent engagement and advancing the conversation around ethical and auditable AI.

9 Whater will only publish or share correspondence (such as emails or feedback) sent by community members if they explicitly agree to this. For example, by including in their communication a statement such as: “You have my permission to publish this response publicly.”

Acknowledgements

This manuscript reports independent research motivated by a commitment to ethical AI governance. Technical development was guided by personal inquiry and iterative problem-solving, aided by AI tools.

Ethical foundations incorporated include Asimov's Laws and Robert Greene's principles manifest the author's commitment to embedding real-world philosophical foundations.

The AI assistant contributed to initial ideation, drafting and subsequently, the structural design of the manuscript. All content has been thoroughly reviewed and refined by the author with AI input to uphold scholarly standards and integrity.

References

- Thatcher, J. (2025). Embedding Reflective Intelligence in AI: A Prototype Implementation of the Wisdom Layer. *SSRN Electronic Journal*. Posted May 28, 2025. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5272939 [1]
- Lewis, P. R., & Sarkadi, Ş. (2024). Reflective Artificial Intelligence. *Minds and Machines*, 34(2), 1-24. 10.1007/s11023-024-09664-2 https://doi.org/10.1007/s11023-024-09664-2 [2]
- Agarwal, A., & Nene, M. J. (2025). A five-layer framework for AI governance: integrating regulation, standards, and certification. *Transforming Government: People, Process and Policy*. Advance online publication. https://doi.org/10.1108/TG-03-2025-0065 . [3]
- (or direct link: https://www.emerald.com/tg/article-abstract/doi/10.1108/TG-03-2025-0065/1256334/A-five-layer-framework-for-AI-governance?redirectedFrom=PDF)
- Salmani, P., & Lewis, P. R. (2025). Adding reflective governance to LLMs. In *Proceedings of the AAAI 2025 Workshop on AI Governance: Alignment, Morality, and Law*. OpenReview. https://openreview.net/pdf?id=qTTlZI0AP6 [4]

Appendix A for Automation and Implementation Patterns

The materials, workflows, and code descriptions provided in this appendix are for illustrative and academic purposes only. Core automation scripts, implementation code, and other proprietary processes essential to the operation of the EveR Self-Training Layer are not included in this public release. These critical assets are protected as intellectual property, with full source code and detailed automation routines available to qualified partners and reviewers under appropriate confidentiality and non-disclosure agreements.

For IP protection and future commercialization, detailed automation scripts and implementation code for core EveR mechanisms are not included in this public release. Reproduction, redistribution, or reuse of any excerpts from this appendix for commercial purposes, derivative work, or integration into other toolchains is strictly prohibited without prior written permission of the author.

For inquiries regarding access to full automation scripts or partnership opportunities, please contact the author at [ben@whater.org and register online at www.whater.org].

Additionally, the project team is actively engaged in a process of identifying and classifying the components of the EveR system that constitute intellectual property (IP) versus those that can be responsibly contributed to the open-source community. Our goal is to maximize the broader impact of the project by sharing valuable non-proprietary elements for the benefit of global developer and user communities, while strategically retaining key innovations and technologies as protected assets. This balanced, ongoing approach is intended to create durable value for project stakeholders and investors, ensure responsible stewardship, and ultimately enable sustainable growth and meaningful benefits across as many communities as possible.

Robust IP safeguards, including planned licensing enforcement with tamper-protection and NDA-gated access, protect EveR's automation code while paving the way for ethical commercialization.

All code, automation scripts, and implementation details marked as proprietary are available only upon request and subject to appropriate non-disclosure arrangements.

Section: Code Automation Patterns and Implementation for EveR

1. Automation Workflow Overview

Purpose and Design of EveR’s Automation Layer

EveR’s automation layer serves as a reflective governance mechanism, designed to systematically prompt the AI assistant to evaluate its outputs and operational behaviour against the principles and criteria set forth in the Living Document. The purpose of this automation is to establish a robust, local governance protocol that moderates AI actions in alignment with user-defined preferences, organizational policies, and ethical frameworks - completely independent of the central AI model's programming.

This framework employs a dual-pathway approach. Human-in-the-loop review (Option A) ensures that all substantive or high-impact changes to the Living Document are explicitly approved by the user, while fully automated process improvement (Option B) facilitates the seamless application of pre-approved, low-risk updates for operational efficiency. All actions, document updates, and audit logs are executed exclusively within the user’s device, thereby supporting strict data privacy, transparency, and regulatory compliance.

The automation engine performs continuous integrity checks to verify the presence and consistency of all required policy and governance files. In the event that any core document is missing or corrupted, EveR halts all non-essential operations, logs the incident for audit purposes, and notifies the user to prompt corrective action.

2. Folder Structure and Initialization

The recommended local directory tree to maximize automation and traceability:

text

```
PrinciplesLivingDocs/                # Root directory
├── PrinciplesMaster.odt             # Primary living document
├── versions/                        # Archive of all prior versions, timestamped
├── logs/                            # Audit logs of upgrades and governance events (e.g., upgrade_log.csv or .ods)
├── proposals/                       # Pending and processed upgrade proposals
└── scripts/                         # Automation scripts (Python with LibreOffice UNO bridge)
```

Initialization Script Sample:

A proprietary initialization script (available upon request) performs checks at each session startup to verify the presence and integrity of these directories and essential files. If any core component is missing or found corrupted, the system halts all non-essential operations, logs the incident for audit readiness, and notifies the user.

This mechanism ensures operational integrity, auditability, and prevents workflow disruption.

This code ensures every session checks for core documents, confirms readiness, and protects against accidental workflow errors - a crucial reliability upgrade validated by your lived experience.

3. Proposal Creation, Review, and Approval Logic

Triggering and Drafting Proposals

Proposal upgrades originate from either AI-detected or user-identified triggers—these may arise from new learnings, risks, or best practices during routine operation, chat interactions, or formal reviews. The process is as follows:

- **Draft Proposal:**

  Upon detection of a relevant event:

  - The assistant drafts a structured proposal containing:

    - The exact recommended text for insertion or modification,
    - Target section or clause,
    - Rationale for the change,
    - Timestamp,
    - Initiating actor (user or assistant).

Option A – Human-in-the-Loop Review

- **User Notification:**

  The system notifies the user in real time, presenting clear response options:

  - (A) Approve as drafted
  - (B) Approve with edits
  - (C) Reject/Defer
  - (D) Request further review

User responses—along with timestamps and rationales—are logged for auditability.

- **Approval and Logging:**

  Only upon explicit user approval will the system:

- Apply the change to the main document,
- Record the action, rationale, user, and timestamp in the structured log file (e.g., `upgradelog.csv`),
- Archive the original proposal for traceability.

Option B – Fully Automated Upgrades

- For low-risk or previously authorized change types (e.g., typo corrections or template fixes), the system can bypass explicit approval:
  - All actions (proposed, applied, and rationale) are still summarized and logged,
  - The user receives a post-action notification with the option to review or roll back as needed.

Implementation and Process Safeguards

- **Structured and Local Logging:**

  All proposals, user interactions, approvals, rejections, and document updates are recorded in a local audit log, maintaining a transparent, tamper-evident change history.

- **File Organization:**

  Activities are confined within a clear local directory structure (`PrinciplesLivingDocs` or `/EveR/`), separating `versions` (backups), `logs`, `proposals`, and `scripts` folders for compliance and reproducibility.

- **Version Control:**

  Every update triggers a timestamped backup of the current document before changes are applied.

- **Error Handling:**

  Any script error halts the process, prompts the user, and prevents unlogged changes, ensuring operational integrity and data protection.

- **Extensibility:**

  Approval workflows and automation thresholds can be adapted as user confidence grows, or audit requirements evolve.

Implementation Note:

Proprietary automation scripts (e.g., Python with UNO for LibreOffice) are available upon request; full code and interface logic are withheld for IP reasons but can be reviewed under NDA. Pseudocode and comprehensive guides for file handling, logging, and user prompts are included in technical appendices.

4. Document Update and Logging Using Python UNO (LibreOffice Automation)

All user-approved or auto-authorized upgrades are enacted using local automation scripts (Python UNO bridge for LibreOffice). These scripts:

- Locate the specified insertion point (e.g., “after Checklist 11”) in *PrinciplesMaster.odt*
- Insert or update recommended text, and save a new document version in the `/versions/` folder for rollback/audit
- Append a structured record (timestamp, action, rationale, user, proposed vs final text) to the upgrade log (`upgradelog.csv`)

Best Practices:

- Test all scripts on backup copies before production use
- Maintain versioned backups of every material change
- Retain local-only storage for legal, privacy, and compliance requirements
- Manual update instructions are available as fallback for first deployments

Implementation Note:

Core automation routines and detailed sample code are available to qualified users and reviewers upon request, subject to NDA and IP protection.

5. Logging and Versioning

The logging and versioning processes are managed by a proprietary automation script (available upon request).

During system initialization, the automation protocol creates the required log templates and directories to ensure comprehensive auditability and traceability.

If any critical log or versioning files are missing or corrupted, the system halts non-essential operations, records the incident in the log, and promptly notifies the user.

This process guarantees consistent and detailed documentation of all governance actions, preserving full historical traceability and supporting regulatory compliance requirements.

6. Security, Testing, and Privacy

All operations within the EveR framework are strictly local to the user's device, ensuring that no data leaves the environment and fully preserving user privacy.

Robust security practices are recommended, including strict folder and file permissions as well as optional operating system-level encryption for protecting data at rest.

Automation scripts are engineered to halt gracefully, safely stopping further operations upon encountering any error in order to prevent data corruption. These scripts are designed to function without requiring elevated administrative privileges.

Summary:

The EveR automation protocol embodies a user-centric, local-first philosophy, enabling scalable, transparent, and auditable continuous improvement for AI governance. The dual-pathway framework includes human oversight (Option A) for significant modifications and automated processing (Option B) for routine, low-risk updates. All changes, whether user-initiated or automated, are meticulously logged with forensic-level detail, facilitating policy evolution, auditability, rollback, and regulatory compliance. This design provides organizations with a privacy-conscious, user-controlled, and fully auditable foundation for responsible adoption of living AI governance practices.

Appendix B for logs

Glossary of Governance Terms and Triggers

This appendix provides concise definitions and references for key governance terms introduced in this paper. For detailed explanations, see the indicated main sections or figures.

Term: The EveR Local Layer: See full definition in blue box on page 4. A local, user-governed reflective governance layer for AI assistants supporting persistent policy enforcement, ethical alignment, and versioned audit control on the user’s device.

The term 'EveR', as used throughout this paper, refers to the full governance framework officially named the 'EveR Local Layer'. This designation emphasizes the platform’s operation as a secure, user-controlled, privacy-preserving layer residing on the user device. When describing the system's structure and function, the phrase 'EveR Reflective Layer' is used to highlight its auditing and alignment capabilities. The earlier terminology 'EveR Self-Training Layer' is retained solely in historical or appendix contexts. This ensures clarity and consistency across the paper and associated documentation.

Term: Use of Phrase “Ps and Qs”
Role: User/system keyword to invoke immediate governance review, policy update proposal, or reflection workflow
Audit: Automatically tagged in all logs
Scope: Available throughout any user-facing interface or automated process, including core documents and personal, non-shared add-ons.

The “Ps and Qs” trigger can be invoked both manually by the user during an interaction and automatically by the system when behavioural inconsistencies, alignment deviations, or operational anomalies are detected, thereby serving as a critical fail-safe and continuous improvement

mechanism. This behaviour is detailed in Sections 3.1 and in the automation scripts overview in Appendix A.

Also see the definitions of Pre-Input Reflection Point and Post-Output Reflection Point triggers in Section 2.3 and annotated in Figure 2.6.1, as integral operational triggers linked to policy alignment and upgrade workflows. See also Sections 3.1 and 4.1 for operational examples.

For full definitions and operational descriptions of governance triggers and audit logs, including the ‘Ps and Qs’ trigger, see Sections 3 (Automation Protocols and Upgrade Workflows) and 4 (Governance Framework and Ethical Integration) of the main text.

References to “Ps and Qs” are consistently incorporated in the operational AI behavioural principles and governance checklists (see Appendix B and relevant sections in the Principles Document), ensuring consistent audit trails and governance alignment across the reflective layer and user's oversight processes.

Appendix C: User Guides

Section: Validation, Testing, and Documentation Best Practice

1. Validation and Acceptance Testing

To ensure EveR’s local automation and living document workflow is robust, secure, and operationally trustworthy, apply the following validation regimes:

- **Startup and Environment Checks**
    - Confirm folder initialization and required core document presence at startup. If critical files (e.g., PrinciplesSelf-training LLM process, AI Behavioural Principles) are missing, the assistant halts and explicitly notifies the user.

- **Functional Flow Testing**
    - Proposal Creation: New proposals appear in the proposals folder, timestamped and structured correctly.
    - Review and Approval: Test all command paths (Approve, Edit, Reject, Defer); all actions must log properly and trigger correct document/log updates.
    - Document Update/Backup: On approval, verify the correct section is updated, backup is saved, and logs capture change, user, and rationale.
    - Logging: After significant actions, confirm the log includes all required traceability fields.
- **Error and Edge Case Handling**
    - Run with missing or locked main document (expect user-friendly errors, nothing unlogged).
    - Test invalid/corrupt proposals (system skips and flags errors).
    - Simulate disk-full/write-access errors (scripts halt cleanly, preserve user data/logs).
    - Reject proposals (logs reflect all administrative decisions).
- **Security and Privacy Validation**
    - All folder/file operations remain local, with no network/cloud interactions.
    - Set restrictive folder permissions (e.g., `chmod -R 700`), verify only the OS user has read/write access.
    - Optionally, verify backup and encrypted restore processes.

---

2. Documentation and Support

Every EveR deployment should include the following core documents:

- **README:** Setup instructions, folder structure, and troubleshooting guidance.
- **Technical Specifications:** Details of file formats, log structures, proposal protocols, and core script workflows.
- **User Guide:** Policy proposal management, backup and recovery procedures, and permissions setup.
- **Versioned ChangeLog:** Tracks all updates to policies and system components for audit and traceability.
- **Security Note:** Outlines the enforcement of local-only operation, password protection, encryption, and recommended backup routines.
- **Master Document ("master_doc"):** The authoritative policy and behavioural reference for the EveR Local Layer. This document should be clearly defined and referenced within the deployment, ensuring all stakeholders and auditors can easily identify the governing source of active policies.

3. Summary Table: Key Test Cases

Test Case	Expected Outcome
Startup w/ missing core doc	Error message, scripts halt; no workflow proceeds
Proposal creation	Proposal file present, correctly structured
Review (all user options)	Each command handled, logs and docs updated as intended
Locked/missing doc	Graceful error, no changes made
Log integrity	Every approval, rejection, and change fully logged
Local-only control	No network activity; all files under local directory

4. Compliance with Best Practice

This validation and documentation suite ensures:

- Reliability and auditability for all living-document actions
- Explicit operational readiness at every session
- Fast debugging and process transparency, matching compliance protocols

With these procedures in place, EveR's local automation is both technically robust and operationally transparent, sustaining a trustworthy and self-improving living documentation environment.

Section: Final Steps and Audit-Ready Project Completion

1. End-to-End Project Checklist and Review

- All core technical documents (design, requirements, specs, plan, guides, validation) delivered
- Folder/naming conventions implemented
- Security/backup protocols in place
- README/guides for collaboration/contributions
- All actions observable and reviewable for audit

This approach aligns your appendices with both the operational best practices and the privacy-first, audit-ready goals of the EveR framework. The documentation and glossary ensure users have clear references for governance triggers and step-by-step guidance for validation, supporting trustworthy, continuous improvement of the living document environment.

Appendix D: Supplementary Diagrams

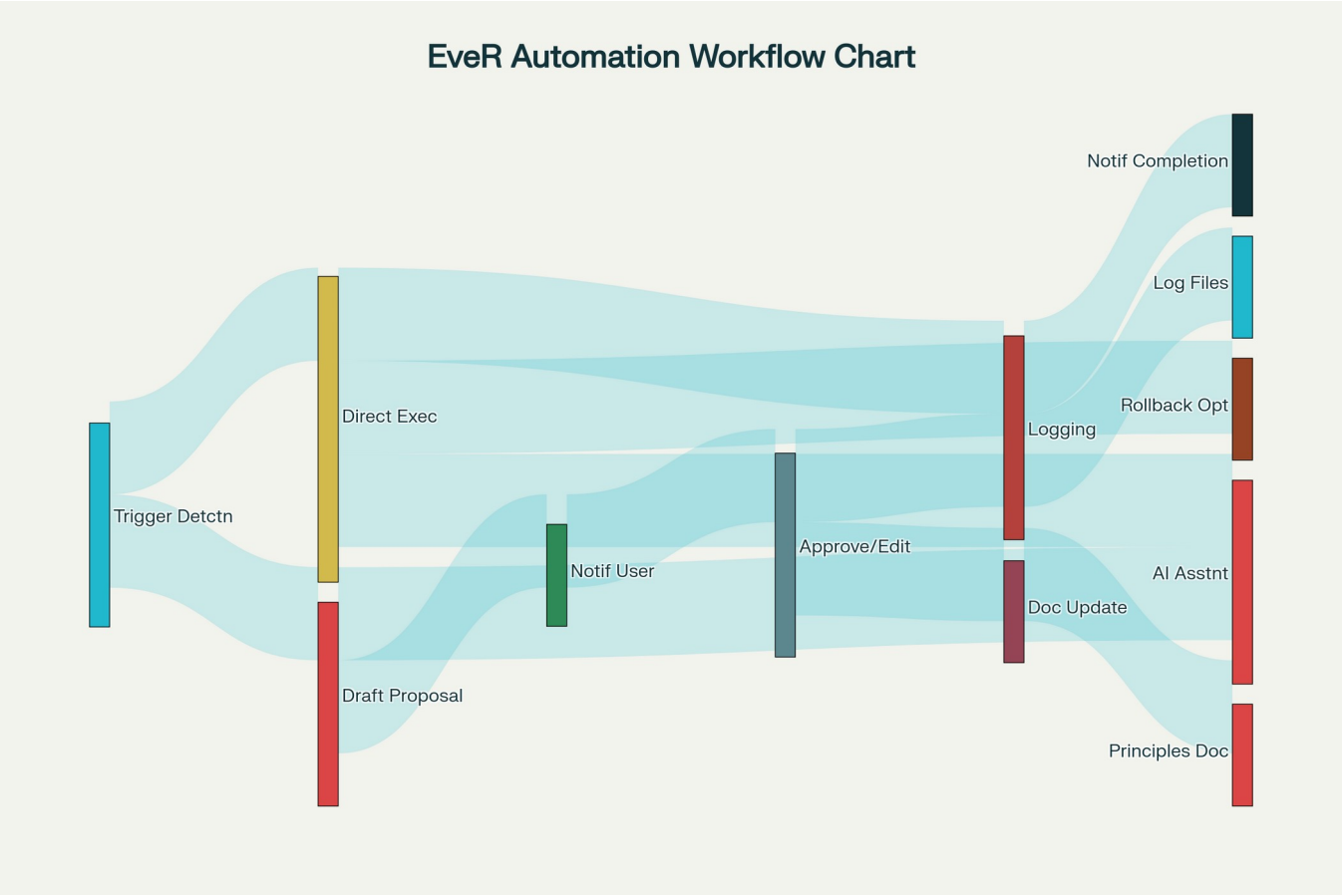


Figure 3.1. EveR Automation Protocols and Upgrade Workflows Diagram

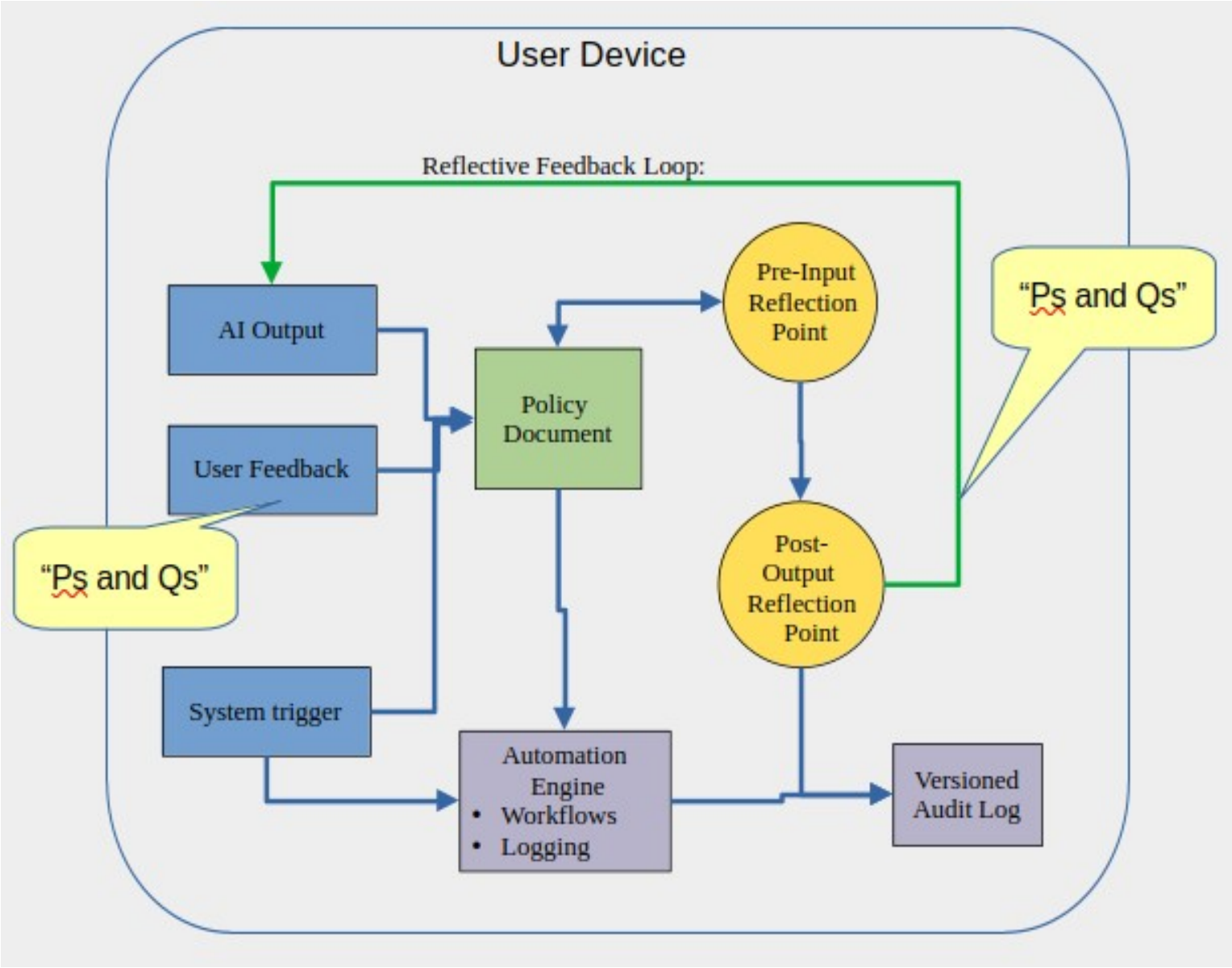


Figure 2.6.1 - Architectural diagram of the EveR Reflective Layer, illustrating key components and data flows within the user device, including input streams, reflection points, automation engine, and versioned audit logs.

Appendix E: EveR Roadmap – Open Ecosystem and Commercial Model

Commercial Pathways and Enterprise Extensions

Building on the foundation of a community-driven ecosystem, the EveR roadmap outlines a comprehensive set of planned business initiatives to support sustainable growth and adoption.

- **Open-Core Model:** The foundational EveR platform and living-document method remain open and freely available for individual, academic, and non-profit use. Certain add-ons that can serve the wider population will forever be free to access.
- **Enterprise Edition (Planned):** A suite of enhanced capabilities including advanced automation protocols, compliance bundles, multi-user governance roles, and priority support aimed at regulated and large-scale deployments.
- **Add-On Marketplace (Planned):** A curated ecosystem for vetted extensions including ethical frameworks, compliance tools, and system integrations, accessible to both community members and enterprise clients. Including governance trigger packs for 'Ps and Qs' customization.
- **Professional Services:** Expert consulting, onboarding assistance, validation services, and certification programs tailored for organizations.
- **Training and Certification:** Accredited workshops and training modules to empower users and organizations with knowledge and capabilities.
- **Governance-as-a-Service:** Hosted and managed governance solutions providing privacy-compliant, regulated oversight for enterprise users.

These pathways ensure EveR’s sustainability, continuous improvement, and broad adoption—balancing open innovation with high-assurance, commercial-ready offerings for professional users.

Appendix F: EveR Roadmap - Open Ecosystem and Commercial Model

Offering	Description	Availability
Open Core	Core self-training and governance features	Current
Enterprise Edition	Enhanced compliance, automation, support	In development
Add-On Marketplace	Curated modules, templates, integrations	Planned
Consulting/Certification	Audit/onboarding/training services	Planned
Governance-as-a-Service	Managed deployment and monitoring	Planned